

0000'

Title FDC MK I CONTROL ROUTINES
subttl Version DS 2.2 13/11/80

0000'

.Z80
ASEG
ORG 0100H
.Comment *

Author: R. Cowderoy

Copyright (C) 1980
R. Cowderoy, P. D. Curtis
and Dondene Ltd.

For use with the Henelec Disk
Controller card with Nascom 1 or 2 or
RICZ80 computer.

Amendment notes:

- 1) Format changed to 11 bytes in L6
for use with Pertek drives. Bytes
sent to disk changed to 0C08H.
PC & RC typed DRH 2/11/80.
Changed to 0C00H DRH 26/11/80
- 2) Conditional assembly for D-DOS
added.
DRH 5/11/80.
- 3) Jumps to LDDRS and LDCMD added to
jump table.
DRH 5/11/80.
- 4) Side select, drive select and
track select separated in LDDRS,
DRSEL and SEEKTR.
PC & RC typed DRH 12/11/80.
- 5) Conditional assembly for RAM at
8800H added.
DRH 16/11/80.
- 6) Format bytes changed to C00H.
DRH 2/12/80

This document is disclosed as a
'Trade secret' supplied for the purposes
of debugging faults within the Henelec
Disk System. Disclosure of part or all of
this document to a third party without
the express permission of the copyright
holders is an infringement of the terms
of the supply agreement. *

```
0100          ; Conditional assembly. Allows assembly for
          ; RICZ80 or NASCOM with either CP/M or
          ; D-DOS V1.0.
FFFF        TRUE   EQU 0FFFFH
0000        FALSE  EQU NOT TRUE

FFFF        NASCOM EQU TRUE
0000        RICZ80 EQU FALSE
FFFF        CPM    EQU TRUE
0000        DDOS   EQU FALSE
0000        RAM    EQU FALSE
FFFF        EPROM  EQU TRUE

          ; PIO port addresses
          IF NASCOM
          ;FDC data port
0004        DPORT  EQU 4
0006        DCNTRL EQU 6
          ; FDC control port
0005        CPORT  EQU 5
0007        CCNTRL EQU 7
          ENDF

          IF RICZ80
          ; FDC data port
          DPORT  EQU 20H
          DCNTRL EQU 21H
          ; FDC control port
          CPORT  EQU 22H
          CCNTRL EQU 23H
          ENDF

00C0        ; Chip mask set to bits 6 & 7 input
          CPMASK EQU 0C0H

          ; Origins of CPM and D-DOS
F400        CPMORG EQU 0F400H
B400        EDOSORG EQU 0B400H
8C00        RDOSORG EQU 08C00H

          ; Data table set up in first 10H bytes defines
          ; the configuration of the system.
          IF CPM
          .PHASE CPMORG
F400    C3 0000    ORIGIN: JP 0
          ; CPM workspace
F403    0048    WSPACE: DEFW 0048H
          ENDF

          IF DDOS AND RAM
          .PHASE RDOSORG-400H
```

```
ORIGIN: JP RDOSORG
; D-DOS workspace
WSPACE: DEFW 0C0EH
ENDIF

IF DDOS AND EPROM
,PHASE EDOSORG-400H
ORIGIN: JP EDOSORG
; D-DOS workspace
WSPACE: DEFW 0C0EH
ENDIF

F405 1000 ISTACK: DEFW 1000H
F407 1000 BOOTST: DEFW 1000H
; Format buffer
F409 1000 FMTBUF: DEFW 1000H
; Maximum number of drives
F40B 03 DRIVES: DEFB 3
; Maximum number of tracks per side
F40C 23 TRACKS: DEFB 35
; Number of retries
F40D 05 NTRY: DEFB 5
IF NASCOM
; Timer delay count
F40E F0 TDEL: DEFB 0F0H
ENDIF
; Timer delay count
IF RICZ80
TDEL: DEFB 90H
ENDIF
; Sides flag, 0 for single, 1 for double
F40F 01 DELS: DEFB 1

; Jump table gives the ONLY legal access points
; to the FDC software
F410 C3 F670 JP READ
F413 C3 F6A5 JP WRITE
F416 C3 F431 JP INIT
F419 C3 F6DA JP FORMAT
F41C C3 F7A1 JP WRBOOT
F41F C3 F573 JP SEEKTR
F422 C3 F604 JP RDENTR
F425 C3 F639 JP WRENTR
F428 C3 F524 JP DRSEL
F42B C3 F507 JP LDDRS
F42E C3 F46A JP LDCMD

; Set data port to input
F431 CD F457 INIT: CALL IMODE
F434 3E FF LD A,0FFH

; Set control port lines to 1
F436 D3 05 OUT (CPORT),A
; Set control port to bit mode
```

```

F438  D3 07          OUT (CCNTRL),A
                    ; Set up control port
F43A  3E C0          LD A,CPMASK
F43C  D3 07          OUT (CCNTRL),A
                    ; Set drive select to 0
F43E  AF            XOR A
F43F  CD F507        CALL LDDRS
                    ; Start motors
F442  CD F4E3        CALL MSTART
                    ; Master reset line to 0
F445  3E 2F          LD A,2FH
F447  D3 05          OUT (CPORT),A
                    ; Delay
F449  06 20          LD B,20H
F44B  05             RSLLOOP: DEC B
F44C  C2 F44B        JP NZ,RSLLOOP
                    ; Master reset line to 1
F44F  3E 3F          LD A,3FH
F451  D3 05          OUT (CPORT),A
                    ; Wait until command done
F453  CD F4C4        CALL WAITBY
F456  C9             RET

F457  F5             ; Subroutine to set data port to bit input mode
IMODE: PUSH AF
                    ; Set to bit mode
F458  3E FF          LD A,0FFH
F45A  D3 06          OUT (DCNTRL),A
                    ; Set all bits to input
F45C  D3 06          OUT (DCNTRL),A
F45E  F1             POP AF
F45F  C9             RET

F460  F5             ; Subroutine to set data port to bit output mode
OMODE: PUSH AF
                    ; Set to bit mode
F461  3E FF          LD A,0FFH
F463  D3 06          OUT (DCNTRL),A
                    ; Set all bits to output
F465  AF            XOR A
F466  D3 06          OUT (DCNTRL),A
F468  F1             POP AF
F469  C9             RET

                    ; Subroutine to load contents of A reg to
                    ; FDC command reg
003C  CMDREG EQU 3CH
F46A  F5             LDCMD: PUSH AF
                    ; Select command reg
F46B  3E 3C          LD A,CMDREG
F46D  D3 05          OUT (CPORT),A
F46F  CD F460        CALL OMODE
F472  F1             POP AF

```

```

; Send command to data port
F473 2F          CPL
F474 D3 04      OUT (DPORT),A

; Pulse write enable
F476 3E 34      LD A,CMDREG-8
F478 D3 05      OUT (CPORT),A
F47A F6 08      OR B
F47C D3 05      OUT (CPORT),A

; Set data port back to input
F47E CD F457    CALL IMODE
F481 C9         RET

; Subroutine to load E reg into FDC sector reg
003E           SECREG EQU 3EH
; Select sector reg
F482 3E 3E      LDSEC: LD A,SECREG
F484 D3 05      OUT (CPORT),A
F486 CD F460    CALL OMODE
F489 7B         LD A,E
F48A 2F         CPL
F48B D3 04      OUT (DPORT),A

; Pulse write enable
F48D 3E 36      LD A,SECREG-8
F48F D3 05      OUT (CPORT),A
F491 F6 08      OR B
F493 D3 05      OUT (CPORT),A
F495 CD F457    CALL IMODE
F498 C9         RET

; Subroutine to read FDC status reg to A reg
003C           STAREG EQU 3CH
; Select status reg
F499 3E 3C      RDSTAT: LD A,STAREG
F49B D3 05      OUT (CPORT),A
F49D CD F457    CALL IMODE

; Read enable to 0
F4A0 3E 38      LD A,STAREG-4
F4A2 D3 05      OUT (CPORT),A

; Read status reg
F4A4 DB 04      IN A,(DPORT)
F4A6 2F         CPL

; Set flags
F4A7 B7         OR A
F4A8 F5         PUSH AF

; Read enable to 1
F4A9 3E 3C      LD A,STAREG
F4AB D3 05      OUT(CPORT),A
F4AD F1         POP AF
F4AE C9         RET

; Subroutine to read FDC track reg to A reg
003D           TRKREG EQU 3DH
```

```

; Select track reg
F4AF 3E 3D RDTRK: LD A,TRKREG
F4B1 D3 05 OUT (CPORT),A
F4B3 CD F457 CALL IMODE

; Read enable to 0
F4B6 3E 39 LD A,TRKREG-4
F4B8 D3 05 OUT (CPORT),A

; Read track reg
F4BA DB 04 IN A,(DPORT)
F4BC 2F CPL
F4BD F5 PUSH AF

; Read enable to 1
F4BE 3E 3D LD A,TRKREG
F4C0 D3 05 OUT (CPORT),A
F4C2 F1 POP AF
F4C3 C9 RET

; Subroutine to check that drive motor is on
; and that busy flag = 0
F4C4 CD F499 WAITBY: CALL ROSTAT
F4C7 F2 F4CD JP P,MOTON

; Start motors
F4CA CD F4E3 MSTRT: CALL MSTART
F4CD CD F499 MOTON: CALL ROSTAT

; Restart motors
F4D0 FA F4CA JP M,MSTRT

; Busy bit to Carry flag
F4D3 1F RRA
F4D4 DA F4CD JP C,MOTON
F4D7 C9 RET

; Subroutine to check if command has timed out
; If so reset FDC using force interrupt
; FDC FORCE INTERRUPT
00D0 FRCINT EQU 0D0H
; Return if top bit = 0
F4D8 B7 TIMEOUT: OR A
F4D9 F0 RET P
F4DA 3E D0 LD A,FRCINT

; Reset FDC busy bit
F4DC CD F46A CALL LDCMD

; Load timeout error code
F4DF 3E 80 LD A,80H
F4E1 B7 OR A
F4E2 C9 RET

; Subroutine to turn on drive motors
; Dummy WRITE to turn on motors
F4E3 CD F482 MSTART: CALL LDSEC
F4E6 E5 PUSH HL

; Delay for 1 second
F4E7 21 03E8 LD HL,1000
F4EA CD F560 CALL MSEC

```

```

F4ED E1 POP HL
F4EE C9 RET

; Subroutine to load A to FDC track reg
F4EF F5 LDTRK: PUSH AF
; Select track reg
F4F0 3E 3D LD A,TRKREG
F4F2 D3 05 OUT (CPORT),A
F4F4 CD F460 CALL OMODE
F4F7 F1 POP AF
F4F8 2F CPL
F4F9 D3 04 OUT (DPORT),A
; Pulse write enable
F4FB 3E 35 LD A,TRKREG-B
F4FD D3 05 OUT (CPORT),A
F4FF 3E 3D LD A,TRKREG
F501 D3 05 OUT (CPORT),A
F503 CD F457 CALL IMODE
F506 C9 RET

; Subroutine to load A to FDC Drive select reg
F507 E5 LDDRS: PUSH HL
F508 2A F403 LD HL,(WSPACE)
; A reg to workspace
F50B 77 LD (HL),A
F50C E1 POP HL
F50D F5 PUSH AF
F50E 3E 3F LD A,3FH
F510 D3 05 OUT (CPORT),A
F512 CD F460 CALL OMODE
F515 F1 POP AF
F516 D3 04 OUT (DPORT),A
; Pulse drive select load
F518 3E 1F LD A,1FH
F51A D3 05 OUT (CPORT),A
F51C 3E 3F LD A,3FH
F51E D3 05 OUT (CPORT),A
F520 CD F457 CALL IMODE
F523 C9 RET

; Subroutine to select drive specified in C reg
; Current drive number stored in WSPACE
; Drive numbers are 0, 2 or 4.
; Subroutine will read new track number from
; the drive
F524 E5 DRSEL: PUSH HL
; Get current drive number
F525 2A F403 LD HL,(WSPACE)
F528 7E LD A,(HL)
; Remove side select bit
F529 E6 FE AND 0FEH
F52B B9 CP C
F52C CA F55D JP Z,SAMEDR

```

```

; Get number of drives
F52F 3A F40B LD A,(DRIVES)
; Correct max drive number
F532 07 RLCA
F533 3D DEC A
; Compare against new drive number
F534 B9 CP C
F535 D2 F53D JP NZ,DNUMOK
; Load drive select fail code
F538 3E 1F SLFAIL: LD A,1FH
F53A B7 OR A
F53B E1 POP HL
F53C C9 RET
; Store new drive number
F53D 79 DNUMOK: LD A,C
; Load drive select reg
F53E CD F507 CALL LDDRS
; 100 µS delay
F541 21 0064 LD HL,100
F544 CD F560 CALL MSEC
; Get address of buffer for read address
F547 2A F403 LD HL,(WSPACE)
F54A 23 INC HL
F54B 23 INC HL
; Load read address command
F54C 3E C0 LD A,0C0H
F54E CD F604 CALL RDENTR
F551 C2 F538 JP NZ,SLFAIL
; Get current track number
F554 2A F403 LD HL,(WSPACE)
F557 23 INC HL
F558 23 INC HL
F559 7E LD A,(HL)
; Load this to FDC track reg
F55A CD F4EF CALL LDTRK
F55D E1 SAMEDR: POP HL
F55E AF XOR A
F55F C9 RET

; Timing subroutine
; Delay number of millisecs specified in HL
F560 C5 MSEC: PUSH BC
; Adjust TDEL for 1 µS
F561 3A F40E MS1: LD A,(TDEL)
F564 47 LD B,A
F565 05 MS2: DEC B
F566 0E 00 LD C,0
F568 C2 F565 JP NZ,MS2
F56B 2B DEC HL
F56C 7C LD A,H
F56D B5 OR L
F56E C2 F561 JP NZ,MS1
F571 C1 POP BC

```

```

F572  C9                RET

; Subroutine to perform seek command to
; track specified in D reg
003F  DATREG EQU 3FH
001F  SEEKCM EQU 1FH
000F  RESTOR EQU 0FH
F573  C5                SEEKTR: PUSH BC
; C reg holds side number
F574  0E 00            LD C,D
; Get number of tracks on disk
F576  3A F40C          LD A,(TRACKS)
; Correct max track number
F579  3D                DEC A
; Compare against requested track
F57A  BA                CP D
F57B  D2 F591          JP NC,SIDE
F57E  47                LD B,A
; Test for double sided drive
F57F  3A F40F          LD A,(DELS)
F582  B7                OR A
; Jump if single sided
F583  CA F5CF          JP Z,TOOBIG
; Calculate new track number for second side
F586  7A                LD A,D
F587  90                SUB B
F588  3D                DEC A
; Check that new number < tracks per side
F589  0E 01            LD C,1
F58B  57                LD D,A
F58C  78                LD A,B
F58D  BA                CP D
; Jump if too big
F58E  DA F5CF          JP C,TOOBIG
F591  E5                SIDE: PUSH HL
F592  2A F403          LD HL,(WSPACE)
; Get drive number
F595  46                LD B,(HL)
F596  E1                POP HL
F597  78                LD A,B
F598  E6 01            AND 1
F59A  B9                CP C
F59B  CA F5AD          JP Z,CPTRK
F59E  78                LD A,B
; Get drive number
F59F  E6 FE            AND 0FEH
; Add side select bit
F5A1  B1                OR C
F5A2  CD F507          CALL LDDRS
F5A5  E5                PUSH HL
; Delay for side select to settle
F5A6  Z1 0001          LD HL,1
F5A9  CD F560          CALL MSEC

```

```

F5AC  E1                POP HL
F5AD  C1                CPTRK: POP BC
F5AE  CD F4AF          CALL RDTRK
F5B1  92                SUB D
                        ; Already on correct track
F5B2  C8                RET Z
                        ; Look for new track
F5B3  CD F5D4          CALL SEEKS
F5B6  C8                ; Return if seek ok
                        RET Z
F5B7  F8                ; Return if drive not ready
                        RET M
                        ; Seek failed, do a reseek
                        ; Restore to track 0
F5B8  3E 0F            RESEEK: LD A,RESTOR
F5BA  CD F46A          CALL LDCMD
                        ; Wait for head to home
F5BD  DB 05            RWAIT: IN A,(CPORT)
F5BF  E6 40            AND 40H
                        ; Wait for IRQ
F5C1  CA F5E0          JP Z,RWAIT
F5C4  CD F499          CALL ROSTAT
F5C7  CD F4D8          CALL TIMOUT
F5CA  F8                RET M
                        ; Try again
F5CB  CD F5D4          CALL SEEKS
F5CE  C9                RET
F5CF  C1                TOOBIG: POP BC
                        ; Load error code for invalid track
F5D0  3E 10            LD A,10H
F5D2  B7                OR A
F5D3  C9                RET
                        ; Seek new track
F5D4  CD F4C4          SEEKS: CALL WAITBY
                        ; Select data reg
F5D7  3E 3F            LD A,DATREG
F5D9  D3 05            OUT (CPORT),A
F5DB  CD F460          CALL OMODE
F5DE  7A                LD A,D
F5DF  2F                CPL
F5E0  D3 04            OUT (DPORT),A
                        ; Pulse write enable
F5E2  3E 37            LD A,DATREG-8
F5E4  D3 05            OUT (CPORT),A
F5E6  F6 08            OR 8
F5E8  D3 05            OUT (CPORT),A
F5EA  CD F457          CALL IMODE
                        ; Send a seek command
F5ED  3E 1F            LD A,SEEKCM
F5EF  CD F46A          CALL LDCMD
F5F2  DB 05            SWAIT: IN A,(CPORT)
                        ; Mask out IRQ input

```

```

F5F4  E6 40          AND 40H
                    ; Wait until IRQ = 1
F5F6  CA F5F2       JP Z,SWAIT
                    ; Read status
F5F9  CD F499       CALL RDSTAT
                    ; Mask error bits
F5FC  E6 98          AND 98H
                    ; Check for timeout
F5FE  CD F4D8       CALL TIMEOUT
F601  C9            RET

                    ; Subroutine to read a sector from the current
                    ; track into memory
                    ; On entry:
                    ;   Sector number = E
                    ;   Buffer start = HL
                    ; On exit:
                    ;   End of buffer = HL
                    ;   FDC status = A
0088          RDCMD EQU 88H
                    ; Read sector comand to A reg
F602  3E 88         RDSECT: LD A,RDCMD
                    ; Entry point for other read commands
F604  C5           RDENTR: PUSH BC
F605  F5           PUSH AF
F606  CD F4C4       CALL WAITBY
F609  06 3F        LD B,DATREG
F60B  0E 3B        LD C,DATREG-4
                    ; Load FDC sector reg
F60D  CD F482       CALL LDSEC
                    ; Retrieve command
F610  F1           POP AF
                    ; Load command to FDC
F611  CD F46A       CALL LDCMD
F614  78           LD A,E
F615  D3 05        OUT (CPORT),A
F617  DB 05        RDWAIT: IN A,(CPORT)
                    ; Mask DRQ and INTRQ
F619  E6 C0        AND 0C0H
                    ; Loop if no requests
F61B  CA F617       JP Z,RDWAIT
                    ; Jump if INTRQ
F61E  F2 F62F      JP P,RDIRQ
F621  79           LD A,C
                    ; FDC read enable to 0
F622  D3 05        OUT (CPORT),A
                    ; Read data byte
F624  DB 04        IN A,(DPORT)
F626  2F           CPL
                    ; Store in buffer
F627  77           LD (HL),A
F628  78           LD A,B
                    ; FDC read enable to 1

```

```

F629 D3 05          OUT (CPORT),A
F62B 23            INC HL
F62C C3 F617       JP RDWAIT
F62F CD F499       RDIRQ: CALL ROSTAT
                   ; Check for timeout
F632 CD F4DB       CALL TIMOUT
F635 C1            POP BC
F636 C9            RET

                   ; Subroutine to write from memory to sector
                   ; on current track
                   ; On entry:
                   ;   Sector number = E
                   ;   Buffer start = HL
                   ; On exit:
                   ;   Buffer finish = HL
                   ;   FDC status = A
00A8 WRCMD EQU 0A8H
                   ; Write sector command to A reg
F637 3E A8         WRSECT: LD A,WRCMD
                   ; Entry point for other write commands
F639 C5            WRENTR: PUSH BC
F63A F5            PUSH AF
F63B CD F4C4       CALL WAITBY
F63E 06 3F         LD B,DATREG
F640 0E 37         LD C,DATREG-8
                   ; Load FDC sector reg
F642 CD F482       CALL LDSEC
                   ; Retrieve command
F645 F1            POP AF
                   ; Load write command to FDC command reg
F646 CD F46A       CALL LDCMD
                   ; Set DPORT to output mode
F649 CD F460       CALL OMODE
F64C 78            LD A,E
                   ; Set CPORT to address FDC data reg
F64D D3 05         OUT (CPORT),A
                   ; Get a data byte
F64F 7E            WBYTEL: LD A,(HL)
F650 23            INC HL
F651 2F            CPL
                   ; Output the data to DPORT
F652 D3 04         OUT (DPORT),A
F654 DB 05         WRWAIT: IN A,(CPORT)
                   ; Mask DRQ and INTRQ
F656 E6 C0         AND 0C0H
                   ; Loop if no requests
F658 CA F654       JP Z,WRWAIT
                   ; Jump if INTRQ
F65B F2 F667       JP P,WRIRQ
                   ; Set FDC write enable to 0
F65E 79            LD A,C
F65F D3 05         OUT (CPORT),A

```

```

; Set FDC write enable to 1
F661 78          LD A,B
F662 D3 05       OUT (CPORT),A
F664 C3 F64F     JP WBYTEL

; Read FDC status
F667 CD F499     WRIRQ: CALL RDSTAT
; Check for timeout
F66A CD F40B     CALL TIMEOUT
F66D C1          POP BC

; Correct buffer pointer
F66E 2E         DEC HL
F66F C9         RET

F670 CD F524     READ:  CALL DRSEL
F673 C0          RET NZ
F674 CD F573     CALL SEEKTR
F677 C0          RET NZ
F67B C5         PUSH BC

; Save buffer start
F679 E5         PUSH HL
F67A 3A F40D     LD A,(NTRY)
F67D 47         LD B,A
F67E 0E 02      LD C,2

; Start of buffer
F680 E1         RFLOOP: POP HL
F681 E5         PUSH HL
F682 CD F602     CALL RDSECT

; Jump if read ok
F685 CA F6A0     JP Z,RDEXIT

; Jump if motor off
F688 FA F6A1     JP M,RFEXIT
F68B 05         DEC B

; Try again
F68C C2 F680     JP NZ,RFLOOP
F68F 0D         DEC C
F690 CA F6A1     JP Z,RFEXIT

; Reset heads to track 0
F693 CD F5B8     CALL RESEEK
F696 C2 F6A1     JP NZ,RFEXIT
F699 3A F40D     LD A,(NTRY)
F69C 47         LD B,A

; Try reading again
F69D C3 F680     JP RFLOOP
F6A0 E3         RDEXIT: EX (SP),HL
F6A1 E1         RFEXIT: POP HL
F6A2 C1         POP BC

; Set flags
F6A3 B7         OR A
F6A4 C9         RET

F6A5 CD F524     WRITE: CALL DRSEL
F6A8 C0          RET NZ
F6A9 CD F573     CALL SEEKTR

```

```

F6AC  C0          RET NZ
F6AD  C5          PUSH BC
F6AE  E5          PUSH HL
F6AF  3A F40D     LD A,(NTRY)
F6B2  47          LD B,A
F6B3  0E 02       LD C,2
F6B5  E1          WFLOOP: POP HL
F6B6  E5          PUSH HL
F6B7  CD F637     CALL WRSECT
                ; Exit if write ok
F6BA  CA F6D5     JP Z,WREXIT
                ; Exit if motor off
F6BD  FA F6D6     JP M,WFEXIT
F6C0  05          DEC B
F6C1  C2 F6B5     JP NZ,WFLOWP
F6C4  0D          DEC C
F6C5  CA F6D6     JP Z,WFEXIT
                ; Restore to track 0
F6C8  CD F588     CALL RESECK
                ; Exit if seek fails
F6CB  C2 F6D6     JP NZ,WFEXIT
F6CE  3A F40D     LD A,(NTRY)
F6D1  47          LD B,A
F6D2  C3 F6B5     JP WFLOWP
F6D5  E3          WREXIT: EX (SP),HL
F6D6  E1          WFEXIT: POP HL
F6D7  C1          POP BC
                ; Set flags
F6D8  B7          OR A
F6D9  C9          RET

                ; Subroutine to format disk
                ; Track addresses from 0 to TRACKS
                ; Sector addresses from 1 to 18

005B          STEPIN EQU 5BH
F6DA  CD F431     FORMAT: CALL INIT
                ; Load initial track address
F6DD  16 00       LD D,0
                ; Load initial sector address
F6DF  1E 01       TRACKL: LD E,1

                ; Set up memory image of track data
F6E1  2A F409     LD HL,(FMTBUF)
                ; 14 bytes to FF
F6E4  3E 0E       LD A,14
F6E6  36 FF       L1: LD (HL),0FFH
F6E8  23          INC HL
F6E9  3D          DEC A
F6EA  C2 F6E6     JP NZ,L1
                ; 6 bytes to 0 for sync
F6ED  3E 06       SECTL: LD A,6
F6EF  36 00       L2: LD (HL),0

```

```

F6F1 23          INC HL
F6F2 3D          DEC A
F6F3 C2 F6EF    JP NZ,L2
                ; ID address mark
F6F6 36 FE      LD (HL),0FEH
F6F8 23          INC HL
                ; Track address
F6F9 72          LD (HL),D
F6FA 23          INC HL
F6FB 36 00      LD (HL),0
F6FD 23          INC HL
                ; Sector address
F6FE 73          LD (HL),E
F6FF 23          INC HL
F700 36 00      LD (HL),0
F702 23          INC HL
                ; ID field CRC
F703 36 F7      LD (HL),0F7H
F705 23          INC HL
                ; 11 bytes to FF
F706 3E 0B      LD A,11
F708 36 FF      L4: LD (HL),0FFH
F70A 23          INC HL
F70B 3D          DEC A
F70C C2 F708    JP NZ,L4
                ; 6 bytes to 0 for sync
F70F 3E 06      LD A,6
F711 36 00      L3: LD (HL),0
F713 23          INC HL
F714 3D          DEC A
F715 C2 F711    JP NZ,L3
                ; Data address mark
F718 36 FB      LD (HL),0FBH
F71A 23          INC HL
                ; 128 bytes data field
F71B 3E 80      LD A,128
F71D 36 E5      L5: LD (HL),0E5H
F71F 23          INC HL
F720 3D          DEC A
F721 C2 F71D    JP NZ,L5
                ; Data field CRC
F724 36 F7      LD (HL),0F7H
F726 23          INC HL
                ; 11 bytes to FF
F727 3E 0B      LD A,11
F729 36 FF      L6: LD (HL),0FFH
F72B 23          INC HL
F72C 3D          DEC A
F72D C2 F729    JP NZ,L6
                ; Increment sector counter
F730 1C          INC E
                ; Test for 18th sector
F731 7B          LD A,E

```

```

F732 FE 13 CP 19
; More to do, so round again
F734 C2 F6ED JP NZ,SECTL
; 255 bytes to FF. End of track gap
F737 3E FF LD A,255
F739 36 FF L7: LD (HL),0FFH
F73B 23 INC HL
F73C 3D DEC A
F73D C2 F739 JP NZ,L7

; Write the data using the write track command
F740 3A F40F LD A,(DELS)
F743 F5 SDLP: PUSH AF
; Side select
F744 CD F507 CALL LDDRS
F747 2A F409 LD HL,(FMTBUF)
; Write entire track command
F74A 3E F4 LD A,0F4H
; Write it
F74C CD F639 CALL WRENTR
F74F B7 OR A
; Return if error in write
F750 C0 RET NZ
F751 E5 PUSH HL
F752 2A F409 LD HL,(FMTBUF)
; Check that 0C00 bytes have been written
F755 01 0C00 LD BC,0C00H
F758 09 ADD HL,BC
F759 4D LD C,L
F75A 44 LD B,H
F75B E1 POP HL
F75C 7D LD A,L
F75D 91 SUB C
F75E 7C LD A,H
F75F 98 SEC A,B
F760 DA F781 JP C,FERR
F763 F1 POP AF
F764 B7 OR A
; Jump if side select = 0
F765 CA F76C JP Z,SDNXT
F768 3D DEC A
F769 C3 F743 JP SDLP
F76C 14 SDNXT: INC D
F76D 3A F40C LD A,(TRACKS)
F770 92 SUB D
F771 C8 RET Z
F772 3E 5B NEXTTR: LD A,STEPIN
F774 CD F46A CALL LDCMD
F777 DB 05 STWAIT: IN A,(CPORT)
F779 E6 40 AND 40H
F77B CA F777 JP Z,STWAIT
F77E C3 F6DF JP TRACKL
F781 F1 FERR: POP AF

```

```

; Format error code
F782 3E FF      LD A,0FFH
F784 C9        RET

; Bootstrap program
; Reads sector 1 of track 0 into memory then
; jumps to it
F785 2A F405   START: LD HL,(ISTACK)
F78B F9        LD SP,HL
; NAG-SYS set up call
F789 CD 000D   CALL 000DH
; Set up drives
F78C CD F431   CALL INIT
; Set drive 0
F78F 0E 00     LD C,0
; Sector 1 track 0
F791 11 0001   LD DE,1
F794 2A F407   LD HL,(BOOTST)
F797 CD F670   CALL READ
; Try again if read fails
F79A C2 F785   JP NZ,START
F79D 2A F407   LD HL,(BOOTST)
; Jump to bootstrap
F7A0 E9        JP (HL)

; Subroutine to write bootstrap to disk
F7A1 CD F431   WRBOOT: CALL INIT
F7A4 0E 00     LD C,0
F7A6 11 0001   LD DE,1
F7A9 2A F407   LD HL,(BOOTST)
F7AC CD F6A5   CALL WRITE
F7AF C9        RET

F7B0 20 43 4F 50   DEFM " COPYRIGHT (C) DONDENE LTD. 1980 "
F7B4 59 52 49 47
F7B8 48 54 20 28
F7BC 43 29 20 44
F7C0 4F 4E 44 45
F7C4 4E 45 20 4C
F7C8 54 44 2E 20
F7CC 31 39 38 30
F7D0 20
```

.DEPHASE

END

Macros:

Symbols:

| | | | | | | | |
|--------|------|--------|------|--------|------|--------|------|
| BOOTST | F407 | CCNTRL | 0007 | CMDREG | 003C | CPM | FFFF |
| CPMASK | 00C0 | CPMORG | F400 | CFORT | 0005 | CPTRK | F5AD |
| DATREG | 003F | DBLS | F40F | DCNTRL | 0006 | DDOS | 0000 |
| DNUMOK | F53D | DPORT | 0004 | DRIVES | F40B | DRSEL | F524 |
| EDOSOR | B400 | EPROM | FFFF | FALSE | 0000 | FERR | F781 |
| FMTBUF | F409 | FORMAT | F6DA | FRCINT | 00D0 | INODE | F457 |
| INIT | F431 | ISTACK | F405 | L1 | F6E6 | L2 | F6EF |
| L3 | F711 | L4 | F708 | L5 | F71D | L6 | F729 |
| L7 | F739 | LDCMD | F46A | LDDRS | F507 | LDSEC | F482 |
| LDTRK | F4EF | MOTON | F4CD | MS1 | F561 | MS2 | F565 |
| MSEC | F560 | MSTART | F4E3 | MSTRT | F4CA | NASCOM | FFFF |
| NEXTTR | F772 | NTRY | F40D | OMODE | F460 | ORIGIN | F400 |
| RAM | 0000 | RDCMD | 0088 | RDENTR | F604 | RDEXIT | F6A0 |
| ROIRQ | F62F | RDOOR | 8C00 | RDSECT | F602 | RDSTAT | F499 |
| ROTRK | F4AF | RDMAIT | F617 | READ | F670 | RESEK | F5B8 |
| RESTOR | 000F | RFEXIT | F6A1 | RFLOOP | F680 | RICZ80 | 0000 |
| RSLOOP | F44B | RWAIT | F58D | SAMEDR | F55D | SDLP | F743 |
| SDNXT | F76C | SECREG | 003E | SECTL | F6ED | SEEKCM | 001F |
| SEEKS | F5D4 | SEEKTR | F573 | SIDE | F591 | SLFAIL | F538 |
| STAREG | 003C | START | F785 | STEPIN | 005B | STWAIT | F777 |
| SWAIT | F5F2 | TDEL | F40E | TIMOUT | F4D8 | TOOBIG | F5CF |
| TRACKL | F6DF | TRACKS | F40C | TRKREG | 003D | TRUE | FFFF |
| WAITEY | F4C4 | WBYTEL | F64F | WFEXIT | F6D6 | WFLOOP | F6B5 |
| WBOOT | F7A1 | WRCMD | 00A8 | WRENTR | F639 | WREXIT | F6D5 |
| WRIRQ | F667 | WRITE | F6A5 | WRSECT | F637 | WRWAIT | F654 |
| WSPACE | F403 | | | | | | |

No Fatal error(s)